Planning Competition for Logistics Robots In Simulation

Scenario and Challenges

Tim Niemueller
Knowledge-based Systems Group, RWTH Aachen University

Erez Karpas
Technion – Israel Institute of Technology

Tiago Vaquero and Eric Timmons
Massachusetts Institute of Technology

The tutorial has received generous support from FESTO

Outline

14:00–14:30 General Introduction and Competition Description

14:30–15:00 Planning and Executive Challenges

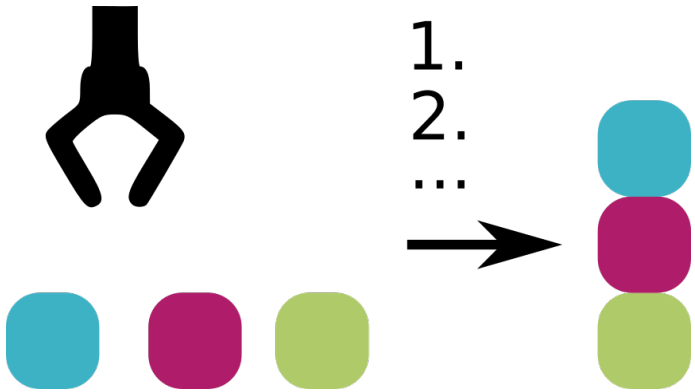
15:00–15:30 Hands-on Part I: Overview and bootup

Coffee Break and Demo

16:00–17:00 Hands-on Part II: Exploring the Simulation

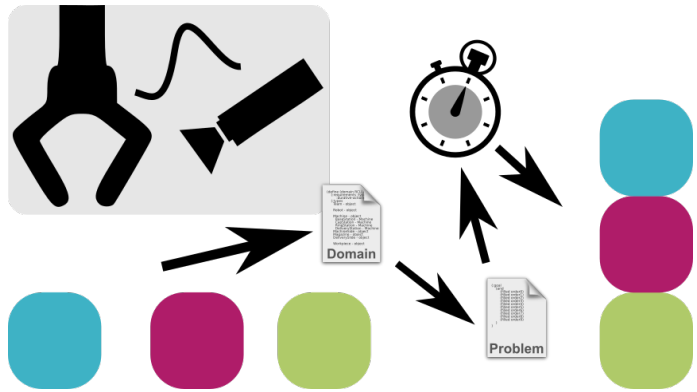
17:00–17:30 Discussion and Feedback

Planning Perspective



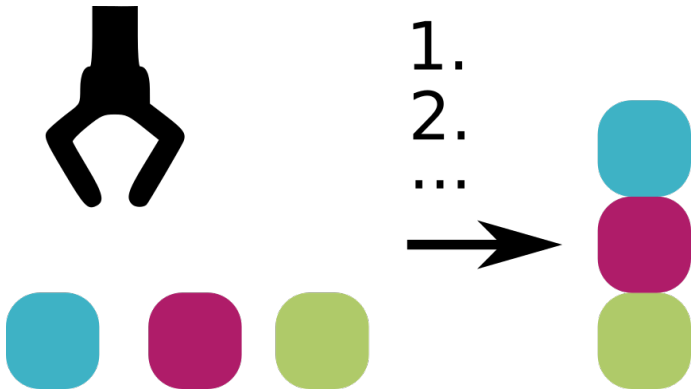
- Focus on plan generation
- Robotics not as testbed
- Execution gets less attention

Robotics Perspective



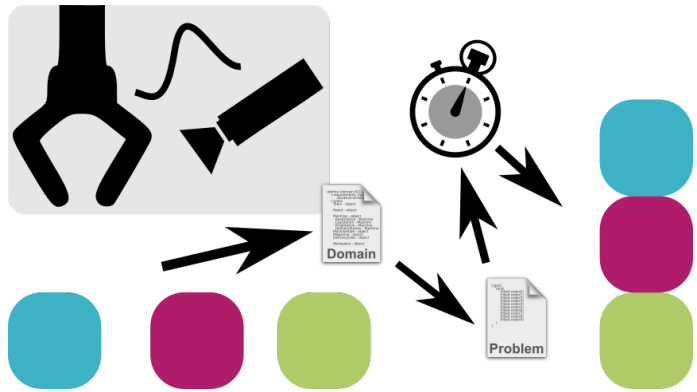
- Focus often on various topics
- Integration for evaluation
- Planning labor-intensive

Planning Perspective



- Focus on plan generation
- Robotics not as testbed
- Execution gets less attention

Robotics Perspective



- Focus often on various topics
- Integration for evaluation
- Planning labor-intensive

Goals

1. Foster closer cooperation among communities
2. Develop grounded expertise with robotic scenarios, platforms, decision architectures, system integration and evaluation

RoboCup Logistics League 2015

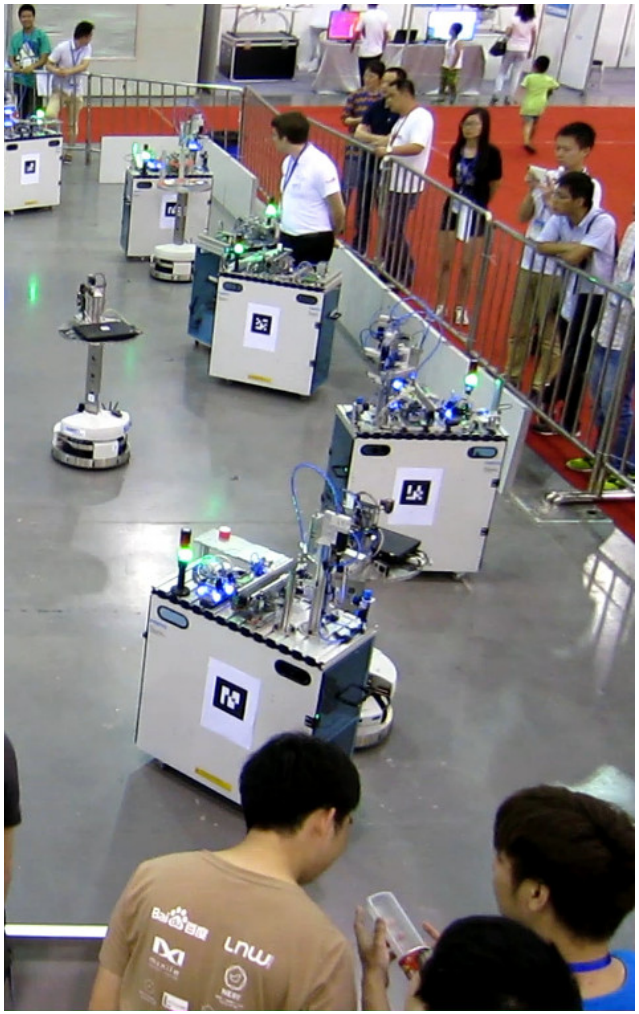
RoboCup Logistics League 2015

Game Basics

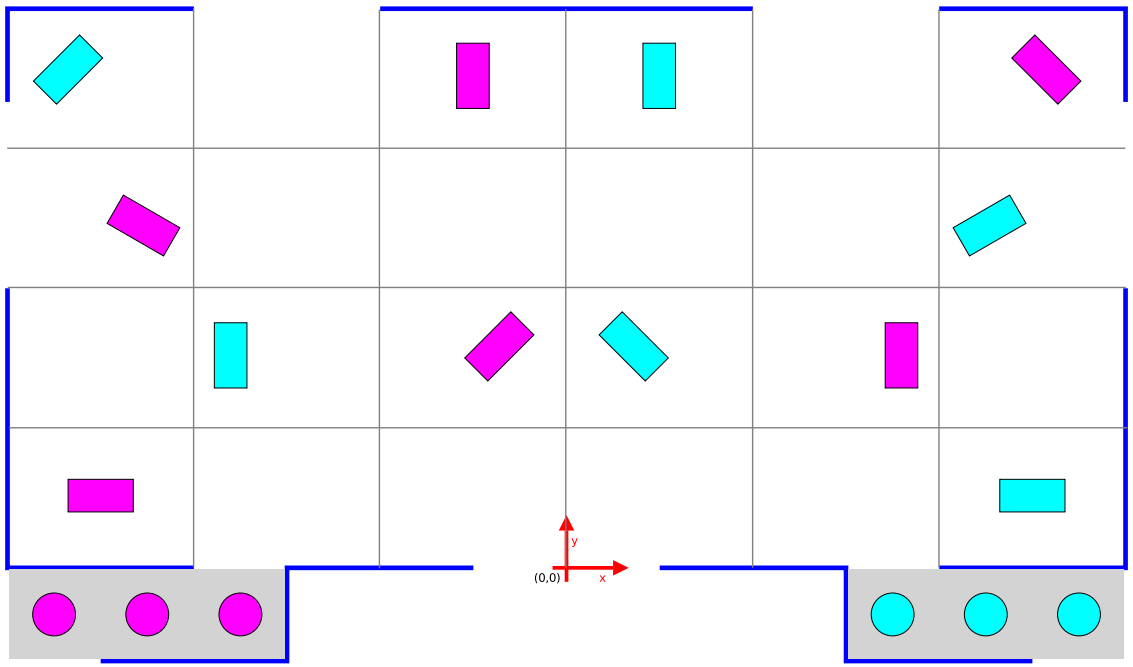
- Task: In-factory production logistics
- Goal: variant production
- Two teams playing on common field
- Each team has 3 robots
- Multi-robot coordination task

Two Game Phases

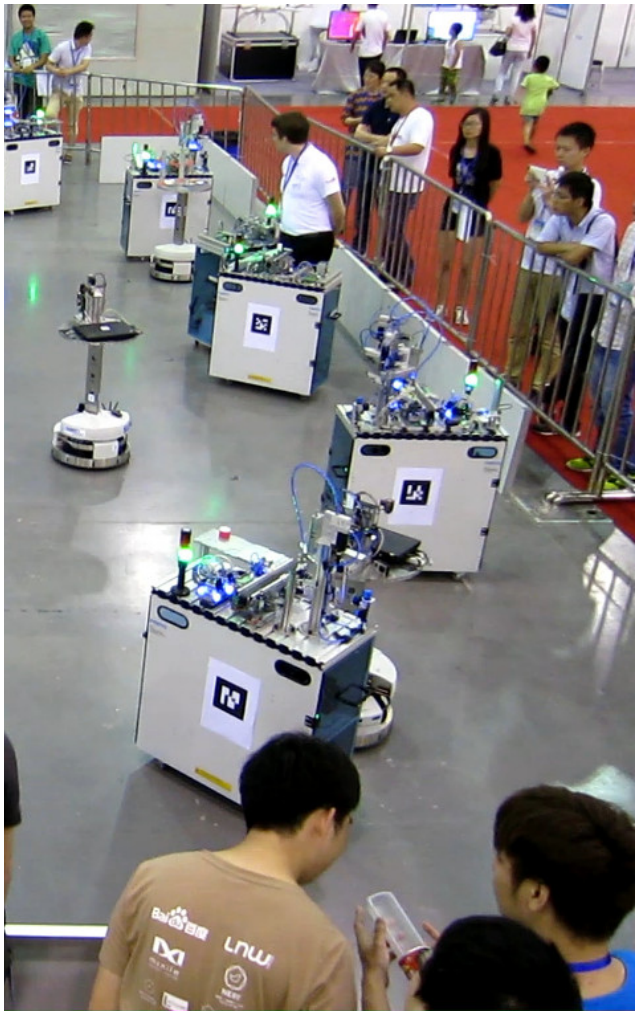
- *Exploration*: detect and report machines
- *Production*: produce and deliver by using processing stations spread across field



Playing Field



- Team colors: cyan and magenta
- Exclusive machines spread across field
- Mirrored at middle axis



RoboCup Logistics League – Machines

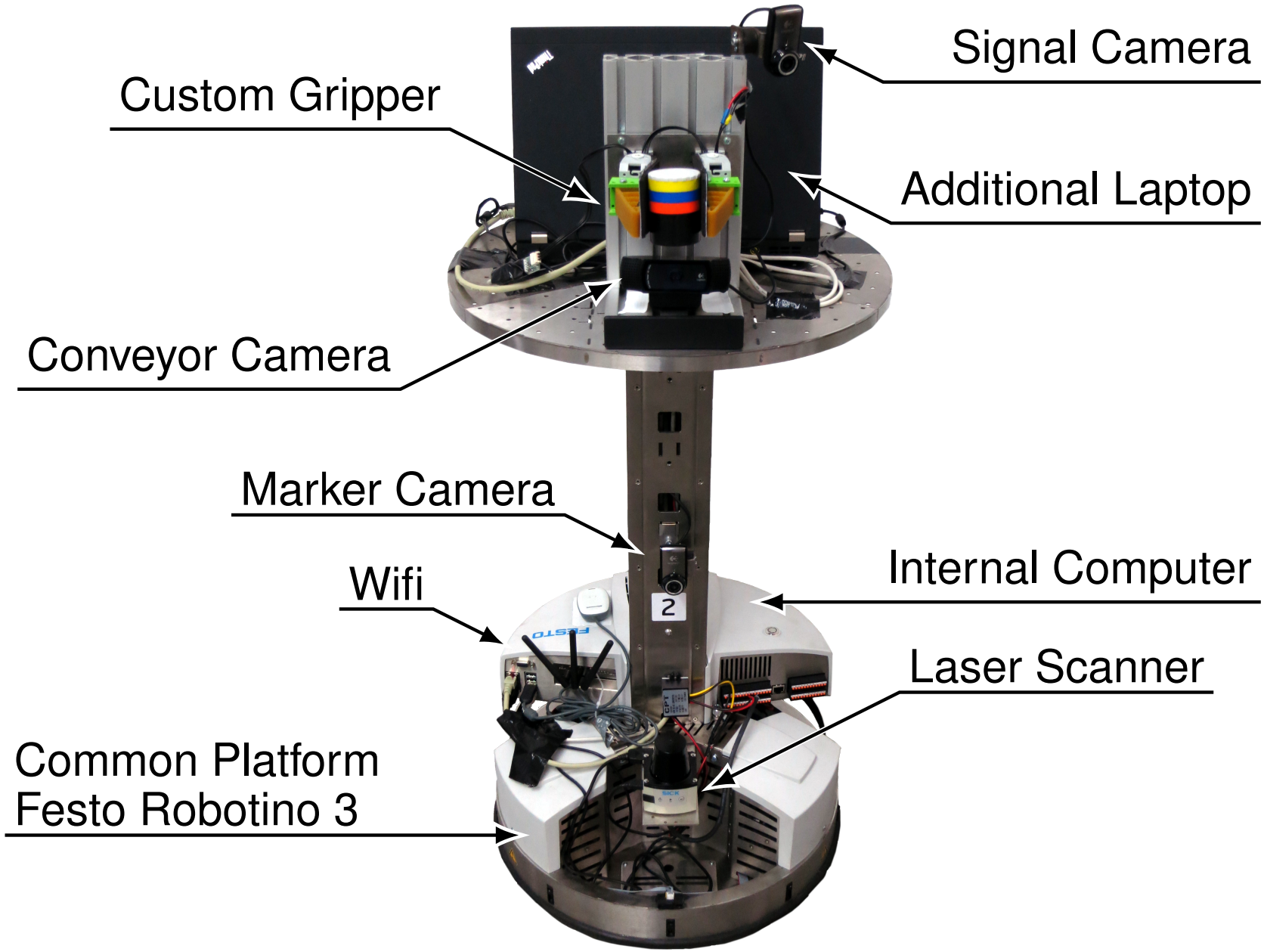
Common

- Based on Festo MPS
- Marker to identify machine
- Signal light to indicate state
- Each team has exclusive set
- Similar handling for all types

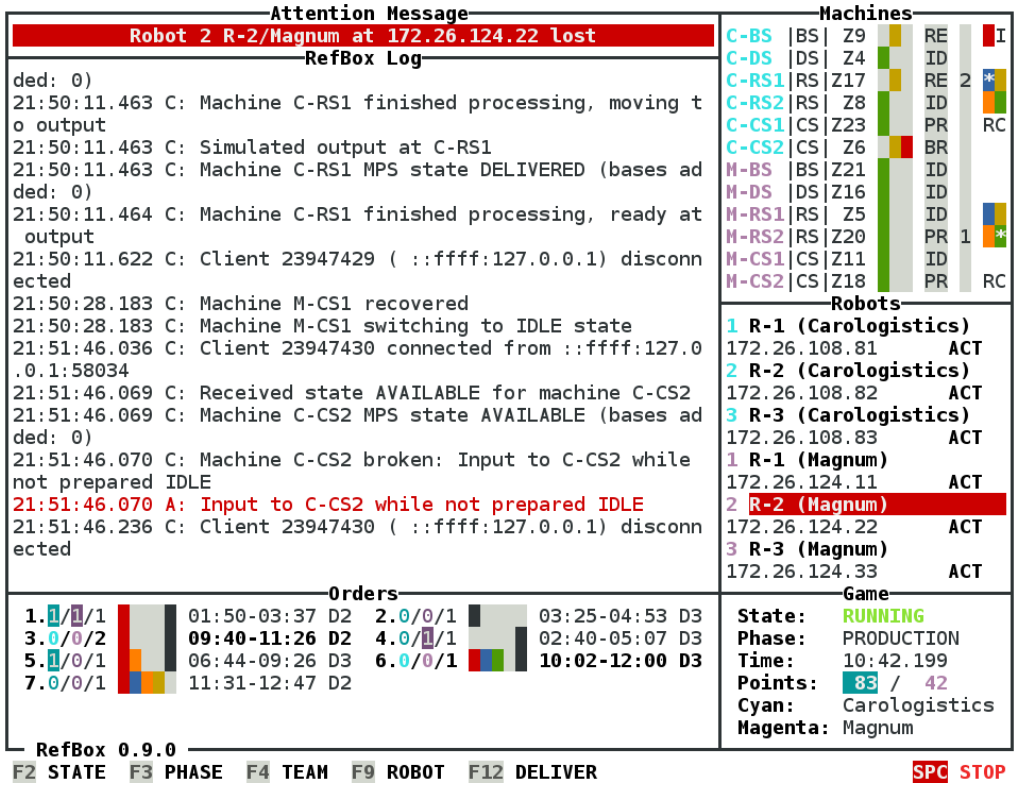
Machine Types (per team)

1× Base Station (BS): retrieve bases
2× Ring Station (RS): mount colored rings
2× Cap Station (CS): buffer/mount caps
1× Delivery Station (DS): final delivery

RCLL Robot Platform (Team Carologistics)



Semi-autonomous Referee Box



Tasks

- Determines randomized orders and machine failures
- Posts orders dynamically
- Scoring and evaluation
- Instructs MPS stations

Planning and Benchmarking

- Accountable environment agency
- Same controller in simulation
- Records extensive data
- Limited uncertainty

$\Rightarrow$ **Repeatable benchmarks**

Logs game information and all communication

Game Phases

Exploration (4 min)

- Machines show light code specific for machine type
- Robot must recognize and announced this type

Production (15 min)

- Orders are posted dynamically, e.g.
  “Deliver 1x P[red base, yellow and green ring, gray cap] in time window [123, 206] to gate 3”
- Robots must complete production chain leading to products
- Coordination is required for effective resource usage
- Machines may fail, other robots on the field

Game Phases

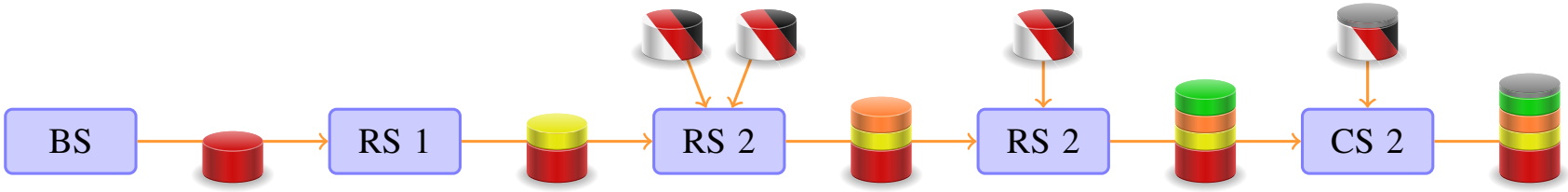
Exploration (4 min)

- Machines show light code specific for machine type
- Robot must recognize and announced this type

Production (15 min)

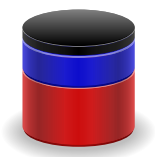
- Orders are posted dynamically, e.g.
  “Deliver 1x P[red base, yellow and green ring, gray cap] in time window [123, 206] to gate 3”
- Robots must complete production chain leading to products
- Coordination is required for effective resource usage
- Machines may fail, other robots on the field

RoboCup Logistics League – Production



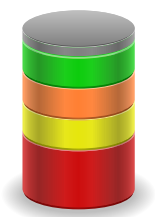
Product Composition

- Products of four complexities (number of rings)
- Base (3 colors) + 0–3 rings (4 colors) + cap (2 colors)
- Order of ring colors is important
- Some ring colors require additional material
- Actual product variants randomized by referee box
- Orders have lead time of a few minutes


Order Elements (posted dynamically by refbox)

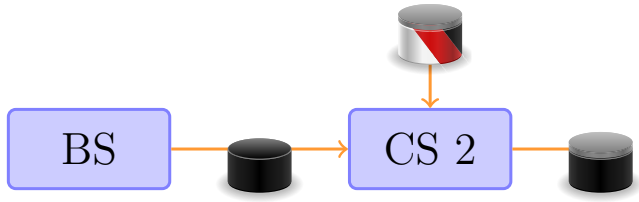
- Product to deliver (and number thereof)
- Time window in which to deliver

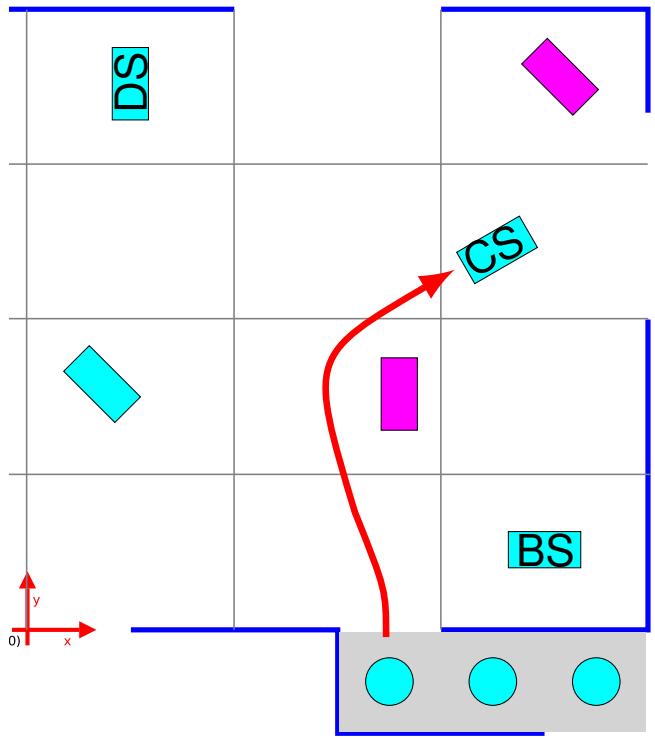


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS

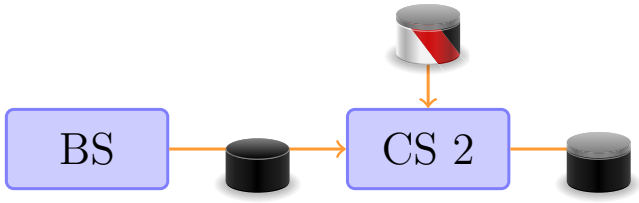


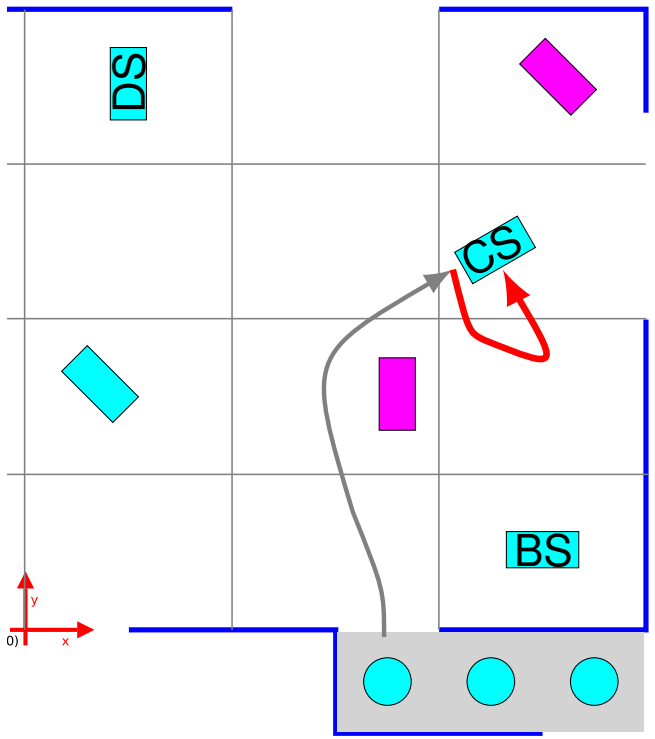


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS

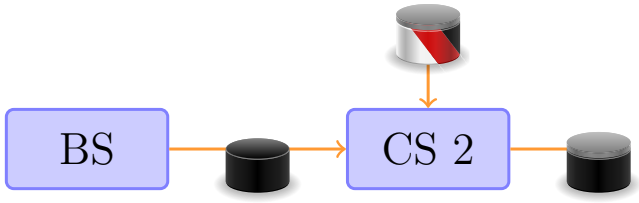


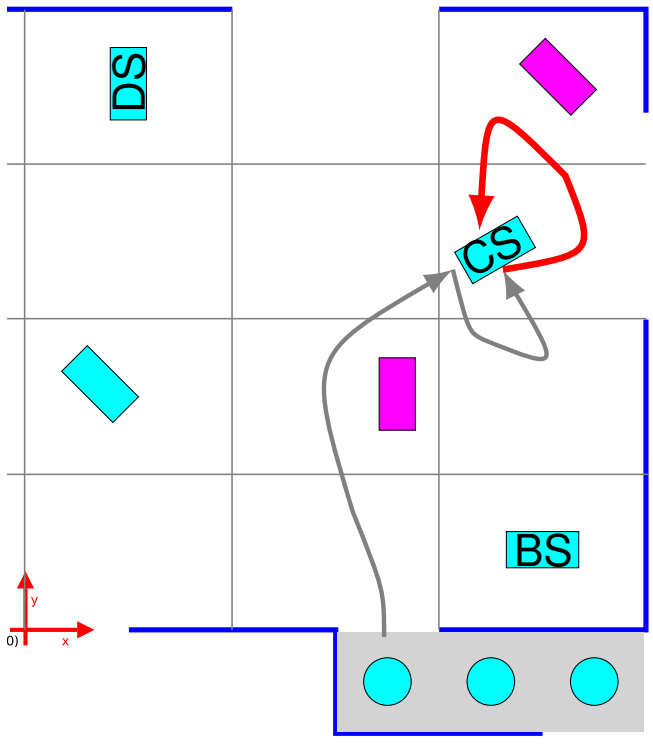


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base

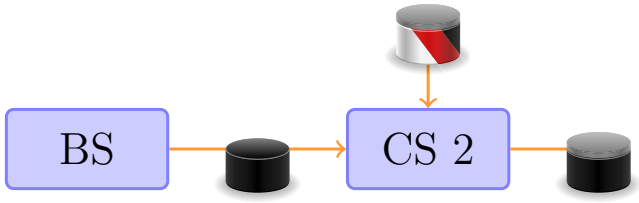


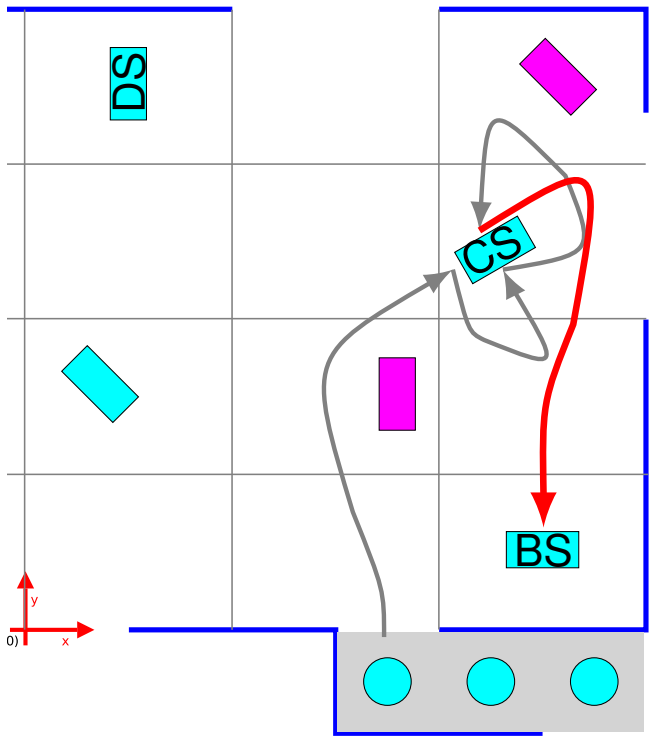


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS

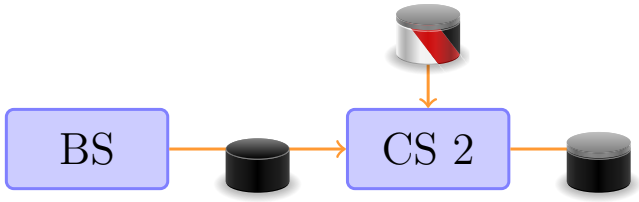


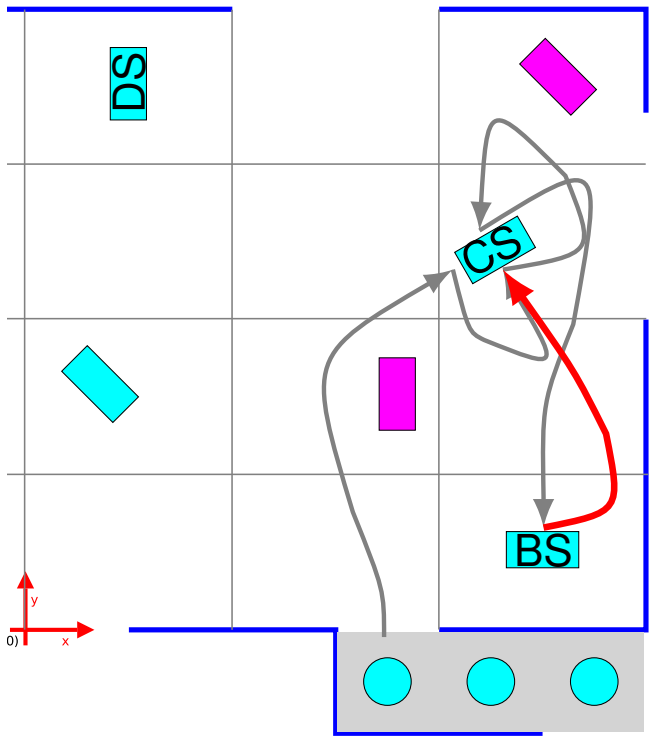


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS

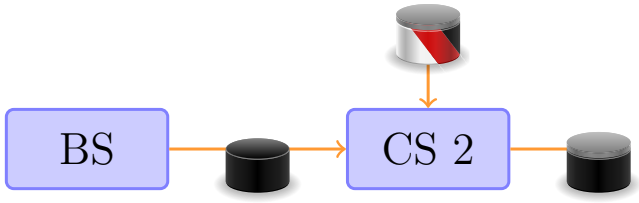


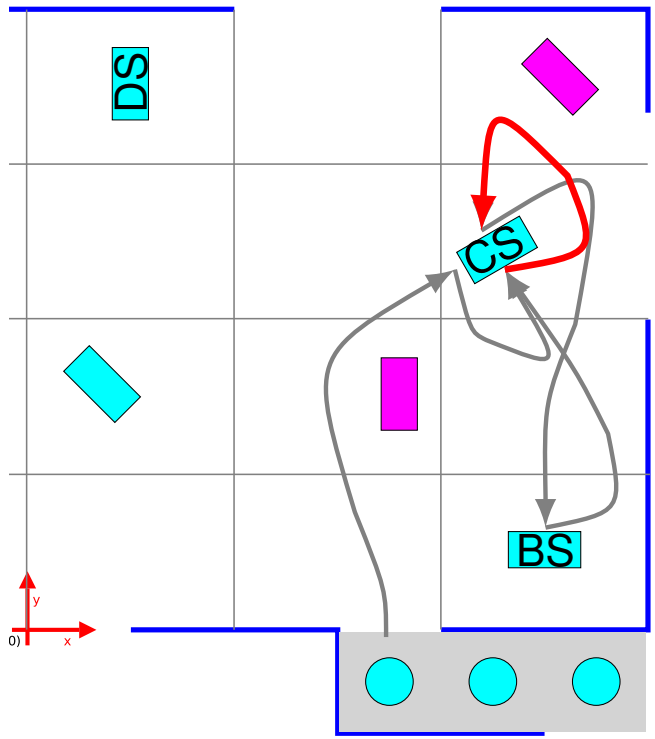


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS

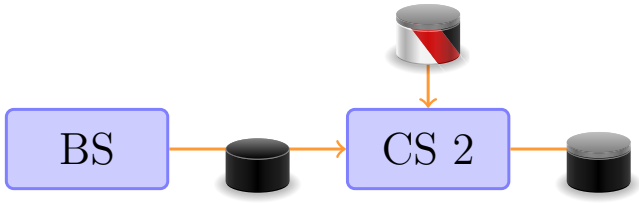


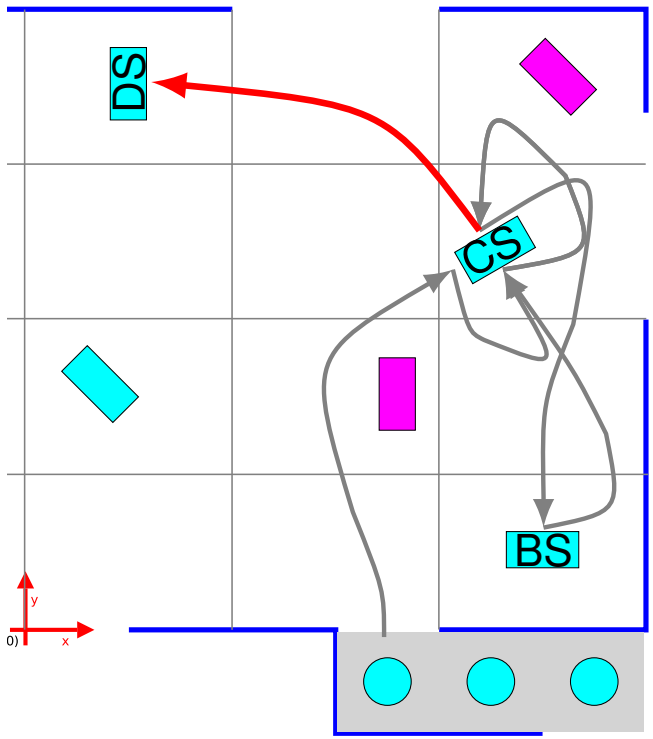


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS

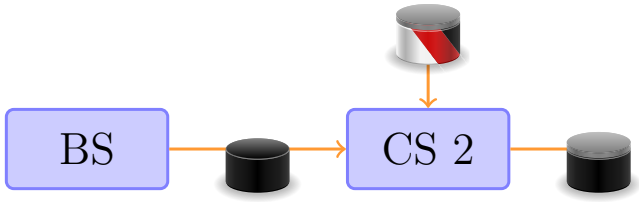


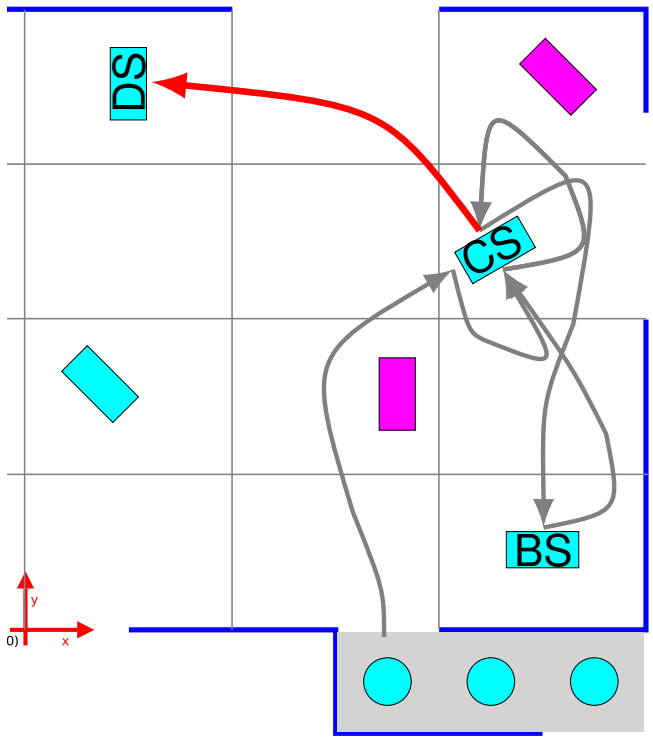


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS





Already simple product has several fragile points and cooperation potential.

Scoring (excerpt)

Sub-task	Description	Points
Additional base	Feed an additional base into an RS	+2
Finish CC_0 step	Finish the work order for a color requiring no additional base	+5
Finish CC_1 step	Finish the work order for a color requiring one additional base	+10
Finish CC_2 step	Finish the work order for a color requiring two additional bases	+20
Finish C_1 pre-cap	Mount the last ring of a C_1 product	+10
Finish C_2 pre-cap	Mount the last ring of a C_2 product	+30
Finish C_3 pre-cap	Mount the last ring of a C_3 product	+80
Mount cap	Mount the cap on a product	+10
Delivery	Deliver one of the final product variants to the designated loading zone at the time specified in the order	+20

Towards a Robotics Planning Competition

RoboCup Logistics League (RCLL)

- In-factory manufacturing logistics in Smart Factory
- Maintain and optimize material flow in production
- Competition under the RoboCup umbrella

RCLL as a Planning Competition and Benchmark

- Cooperative and competitive aspects, partially observable, non-deterministic, dynamic
- Typical: local, distributed, incremental strategy
- Desired: planning for global optimization
- Challenges: coordination, execution, robustness

Towards a Robotics Planning Competition

RoboCup Logistics League (RCLL)

- In-factory manufacturing logistics in Smart Factory
- Maintain and optimize material flow in production
- Competition under the RoboCup umbrella

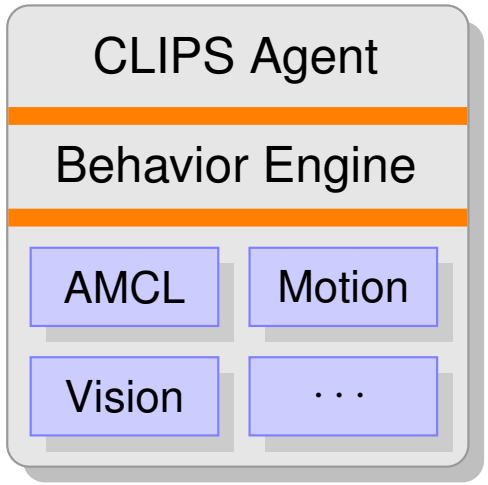
RCLL as a Planning Competition and Benchmark

- Cooperative and competitive aspects, partially observable, non-deterministic, dynamic
- Typical: local, distributed, incremental strategy
- Desired: planning for global optimization
- Challenges: coordination, execution, robustness

Medium complex benchmark domain focusing on efficient *planning/scheduling* and **execution integration**

CLIPS-based Incremental Task-Level Reasoning

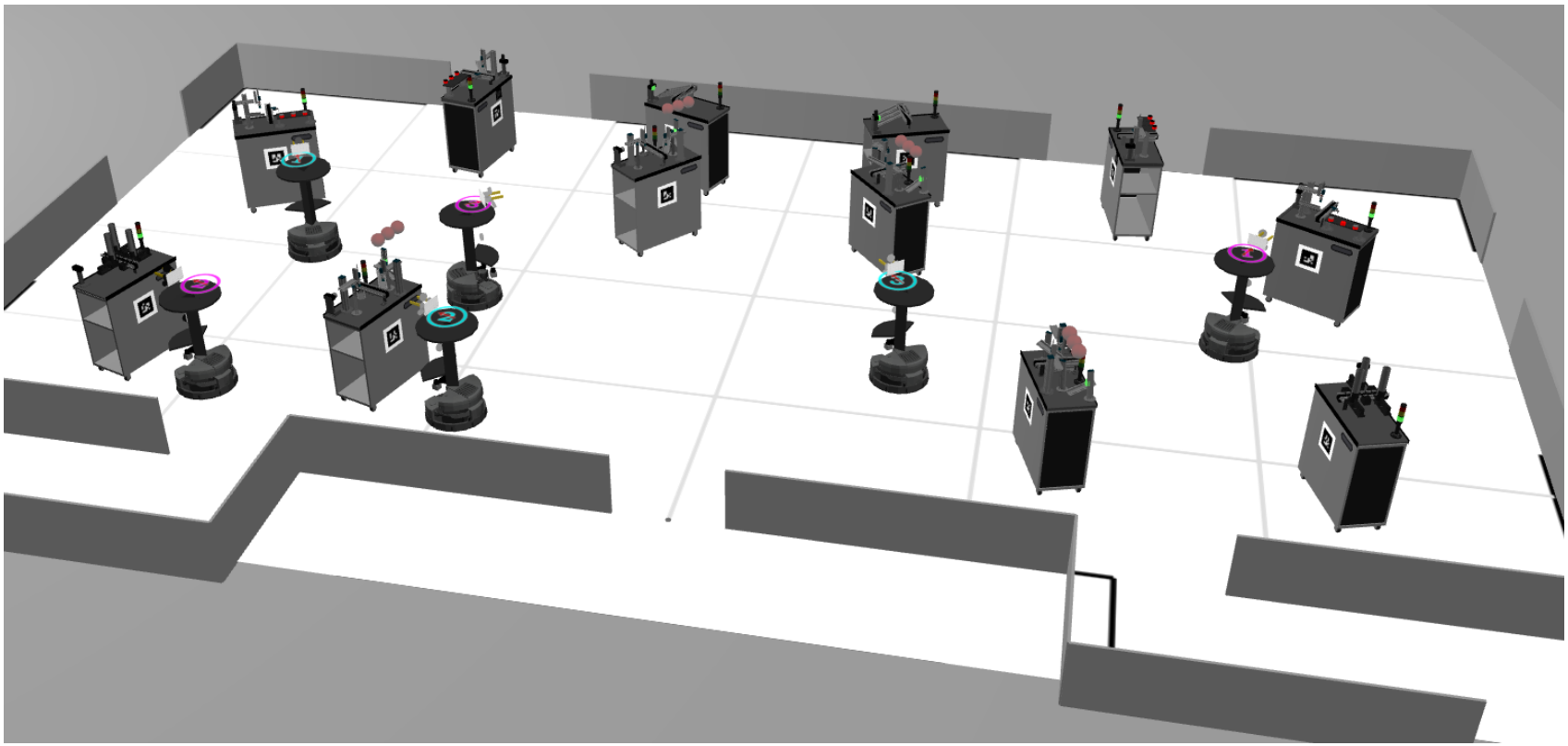
- Only commit to single step at a time
- Strategic behavior with coarse tasks
- Reason about current knowledge



- CLIPS rule-based system
- Efficient reasoning with many updates
- Distributed, local-scope, incremental

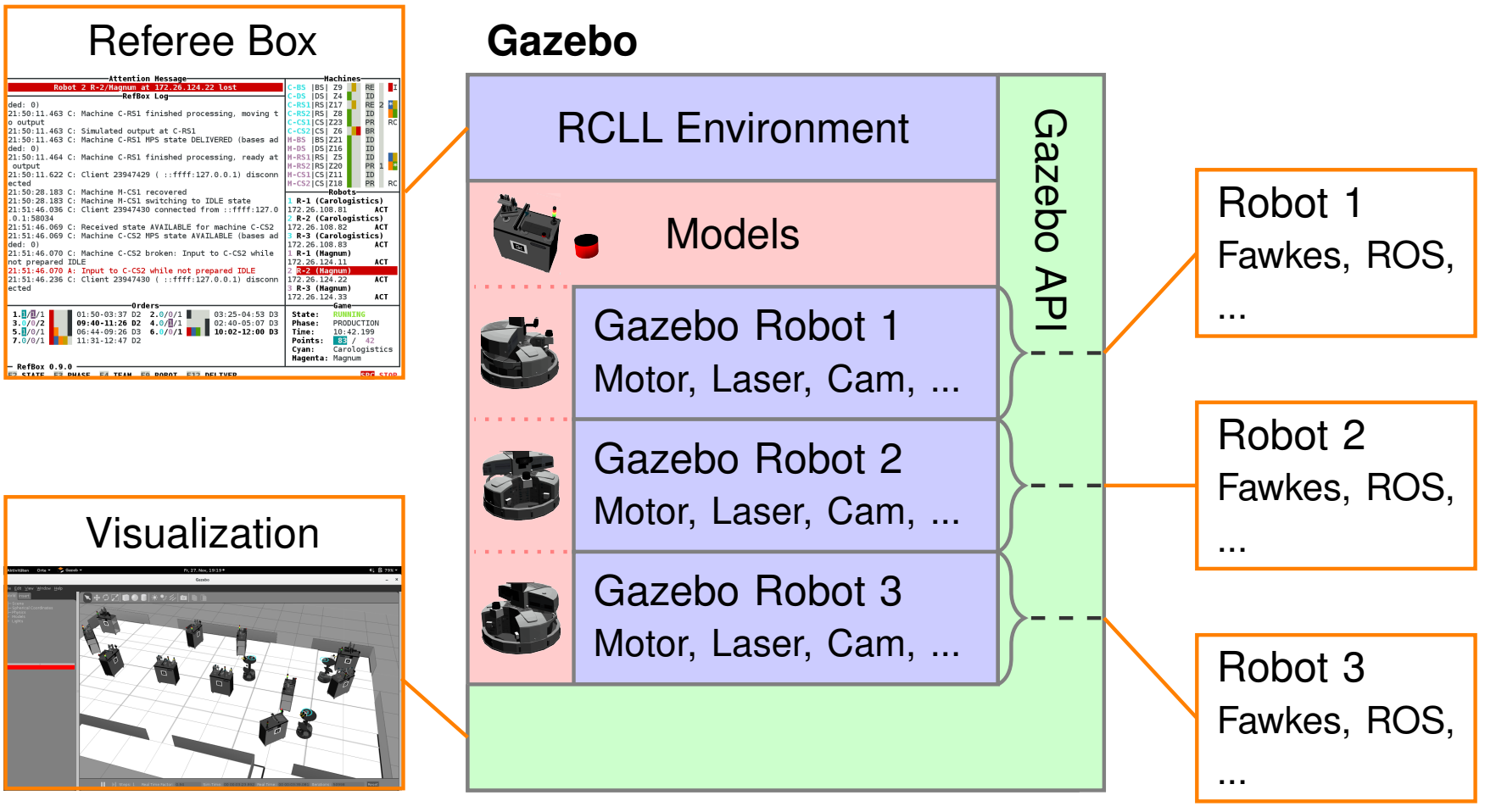
```
(defrule s1-t23-s0
  (state IDLE) (holding S1)
  (machine (mtype ?mt&M2\_3) (name ?n)
           (loaded-with $?lw&:(contains$ S0 ?lw)) )
  =>
  (assert (task-candidate goto ?n))
)
```

RoboCup Logistics League – Simulation



- Readily integrated 3D simulation with environment agency
- Based on software stack by RoboCup Team Carologistics

Simulation Architecture



System Integration

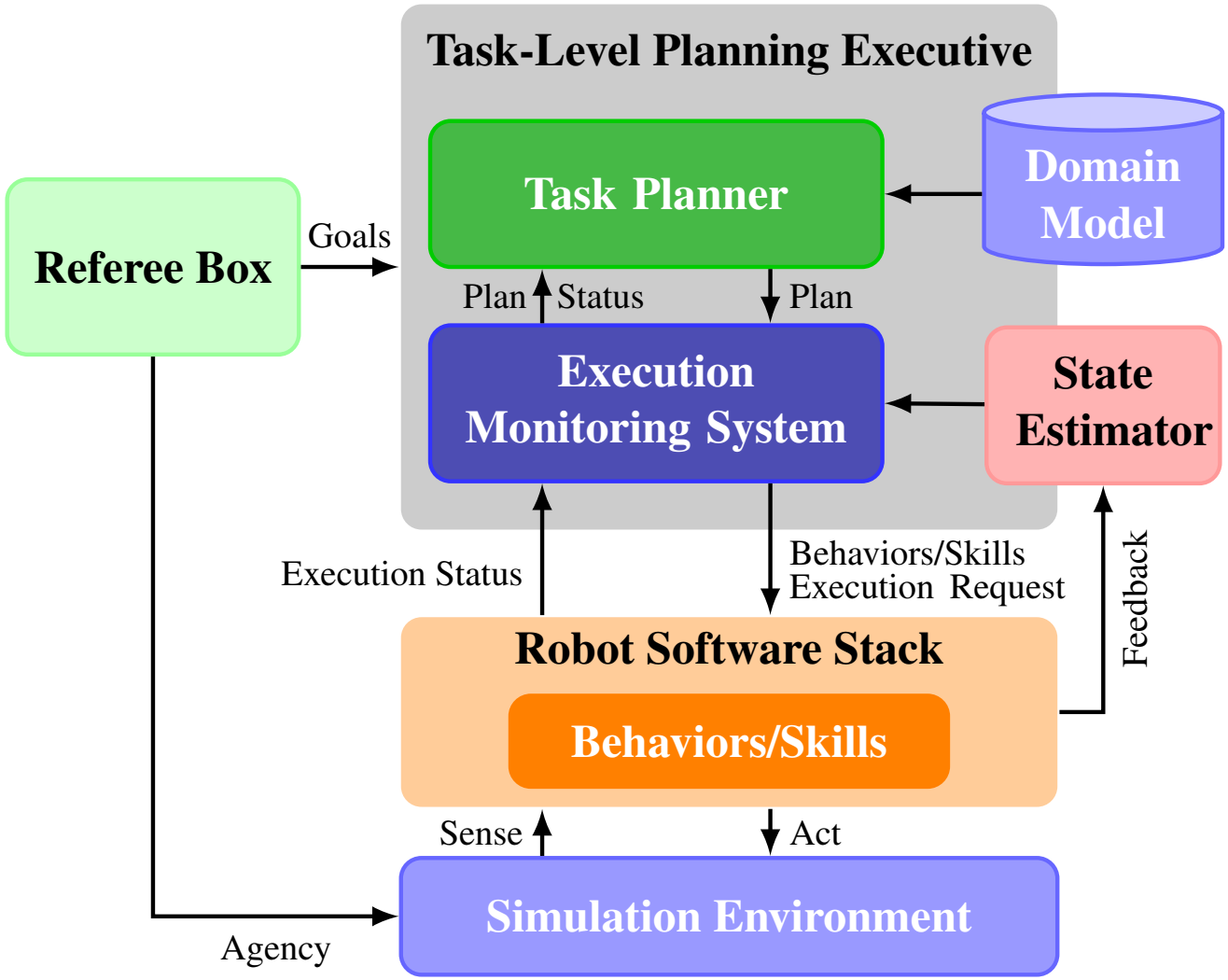
Fawkes Robot Software Framework

- Functional software components
- Lua-based Behavior Engine for skill execution
- Path planning and locomotion

ROS

- Full integration with simulation
- Encapsulates communication with referee box
- Visulization tool

Planning System Architecture



Discussion Proposal: Competition Tracks

Tracks are determined by what can be replaced or extended.

Track 1: Planner and Execution

Domain model, task planner, execution monitoring, state estimator

Track 2: Behaviors and Motion Planning

Skills/behaviors, execution engine, motion planner (local/global)

Track 3: Free style

Any component but the simulation (parts) and its interface.

Planning and Robotics Competition

Challenge

Integrated planning and execution in a medium complex simulated robotics industry-inspired scenario

Tracks

Accommodate diversity by creating several tracks that have their specific extension points (discussion)

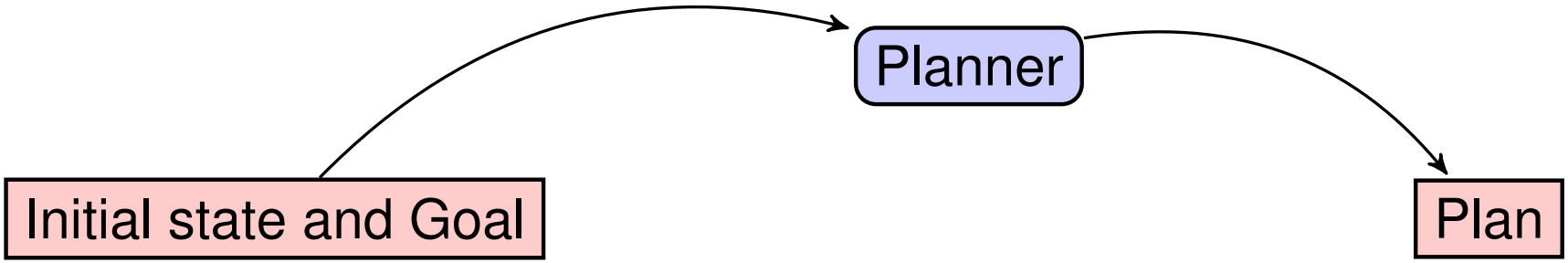
Timeline

2015: Presentation of RoboCup Logistics League scenario
2016: Tutorial, discussion of scenario and tracks
2017: Competition at ICAPS 2017

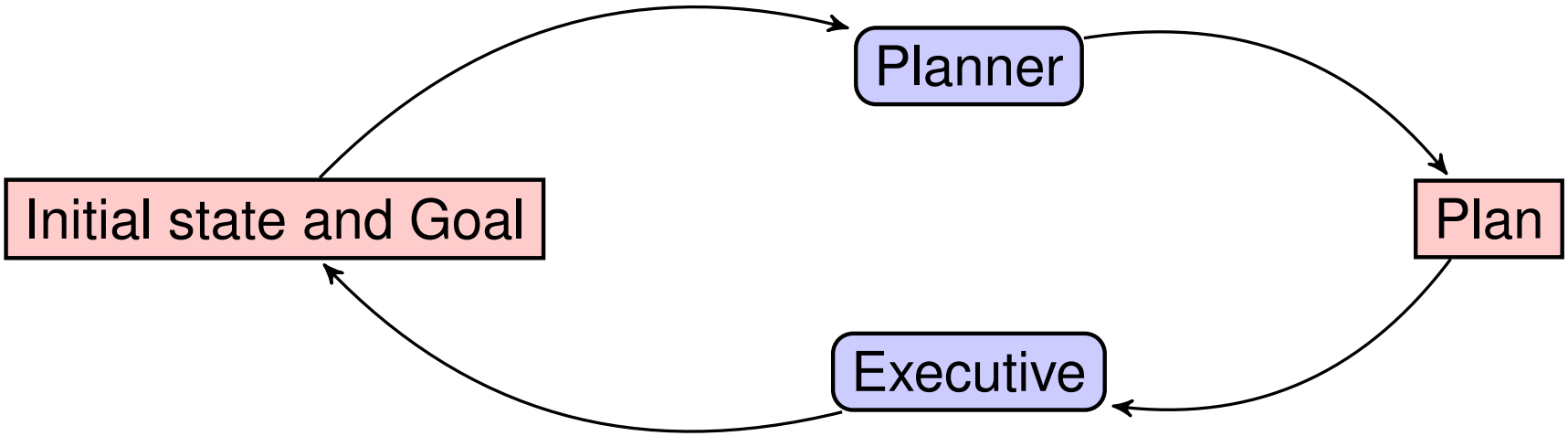
Planning Challenges

Erez Karpas

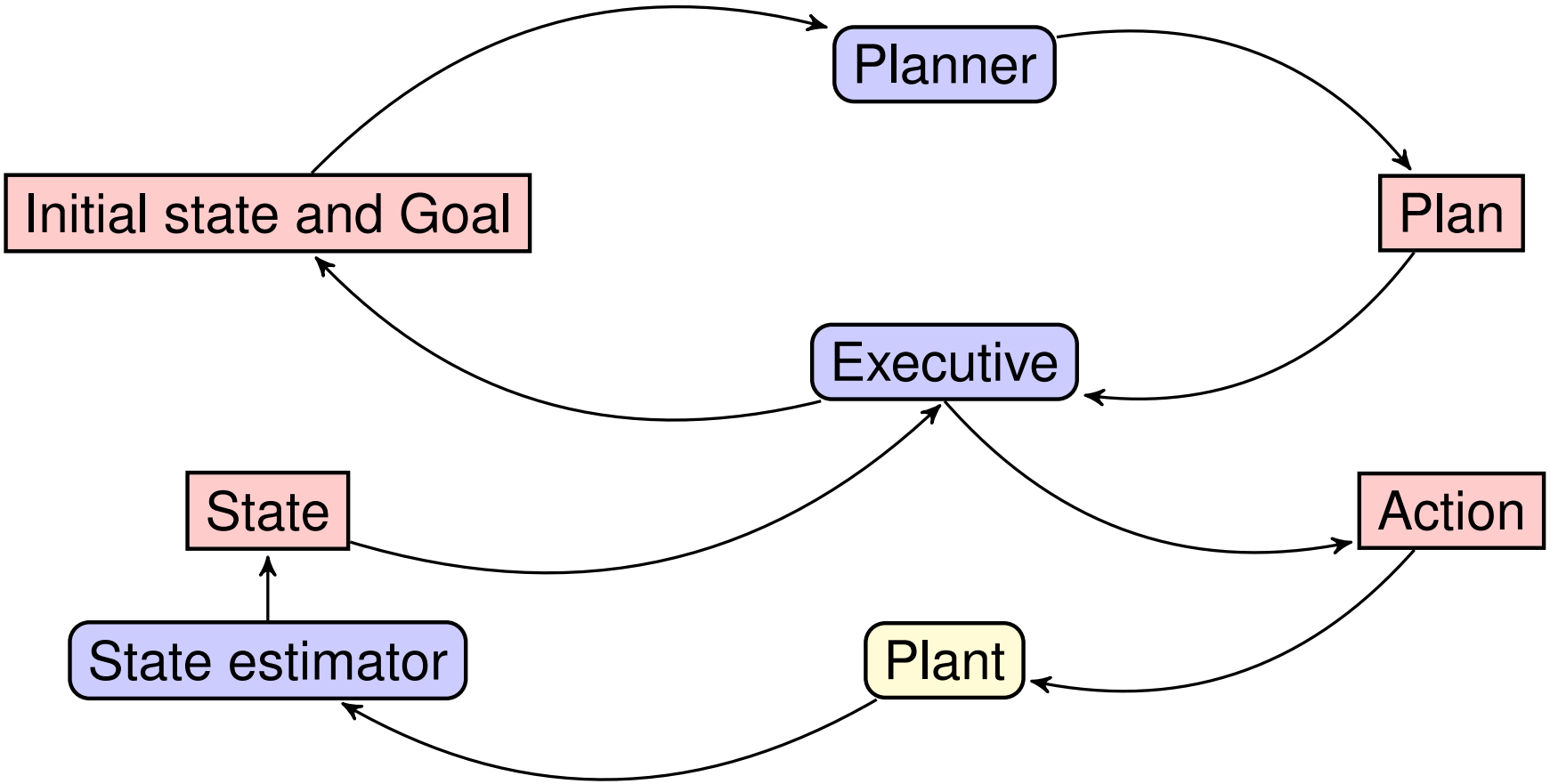
Planning



Planning and Execution



Planning and Execution with Plant



Challenges: Planning

- Finding sequence of actions for each order — easy
- Assigning robots to orders/subtasks
- Large number of objects
  - Our domain formulation keeps track of each workpiece by its identity
  - Including workpieces that are symmetric

Challenges: Planning and Execution

- Orders arrive on the fly
  - Replanning
  - Anticipating orders
- Competition
  - What is the objective?

Challenges: Planning and Execution with Plant

- Actions can:
  - fail (in predicatable and unpredictable ways)
  - take too long (or too little)
- Adjust plan based on opponent/ranking?

Running the Simulation

Starting the Simulation

1. Boot from USB stick
2. Login: robosim/simcomp2017
3. Get laptop on wifi (e.g., eduroam)
4. Run Terminal
5. Update and rebuild simstick

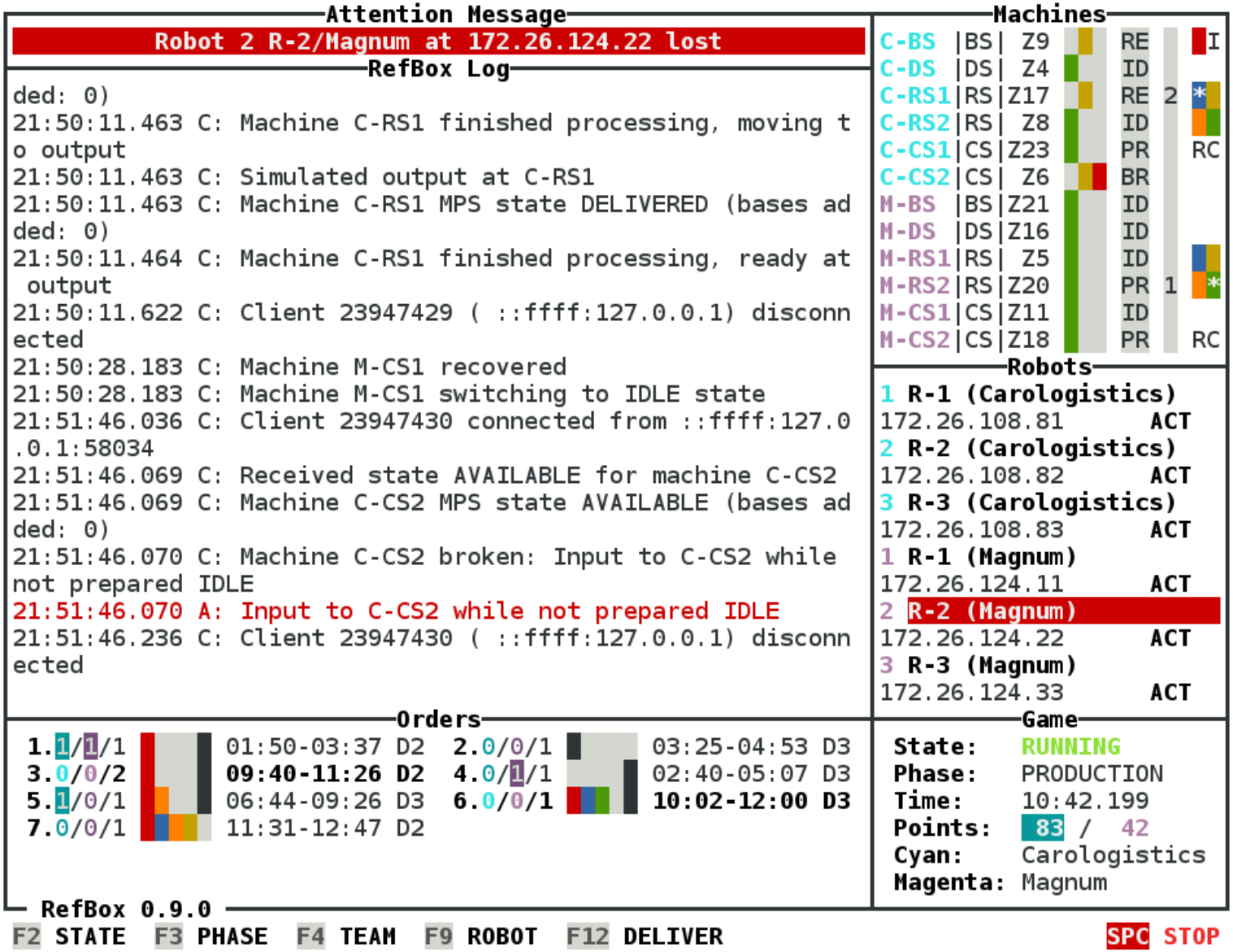
```
# simstick-update
# simstick-rebuild
```

6. Start Simulation

```
# cd robotics/fawkes-robotino/bin
# ./gazsim.bash -x start -r -n 1 -t -a
```

7. In Gazebo, hit F11 twice to get to window mode

Running the Game



Explore ROS Integration

Start Simulation with ROS Integration

```
# cd robotics/fawkes-robotino/bin
# ./gazsim.bash -x start -r -n 1 -t \
  --ros-launch-main \
  rcll_fawkes_sim:rcll_fawkes_sim_all_1robot.launch
```

Note: **no** `-a` flag!

Explore Topics

```
# rostopic list
```

- Per robot namespace, /robot1 etc.
- /robotN/rcll: communication with referee box
- /robotN/rcll_sim: simulation integration
- /robot1/skiller: skill execution action

Explore ROS Integration

Run Skill via ROS Action

```
# rosrun actionlib axclient.py \
    /robot1/skiller fawkes_msgs/ExecSkillAction
```

Send the following goal:

`skillstring: 'ppgoto{place="C-CS1-I"}'`

(requires to be in the production phase in refbox)